We Claim:

1. A method for initializing a Random Access Memory (RAM) having a set of address spaces, comprising the steps of:

detecting a reset event;

disabling status bits of said set of addresses;

over-writing token data, when allocated, sequentially to said address spaces, and updating respective status bits upon each write event; and

when all said address spaces are occupied, enabling said status bit to be active upon address tokens being read from and written to said RAM.

2. A method for initializing a Random Access Memory (RAM) having a set of address spaces for token data and a respective status bit, comprising the steps of:

detecting a reset event;

discarding any allocated token data;

allocating token data to said address spaces sequentially and setting or maintaining a status bit to logic "1" upon each allocation; and

when all said address spaces are occupied, reverting to actual status bit condition for subsequent token data allocations.

3. A Random Access Memory intilialization circuit comprising:

a memory having a set of address spaces to which token data is written to and read from, each address space having a status bit indicating allocation of a token data thereto;

a token generator for allocating token data to said memory address spaces; and

a logic circuit; and

wherein upon a reset event occurring, said logic circuit providing a logic "0" input to said token generator, and said token generator allocating token data to said address spaces sequentially and the respective memory status bit being maintained or set to logic "1" upon each allocation, and further wherein, upon all said address spaces being occupied by allocating token data, said logic circuit

provides the actual state of the status bit of said address spaces to said token generator to control subsequent allocations.

4. The Random Access Memory initialization circuit of claim 3, wherein said logic circuit includes an R-S flip-flop receiving a reset signal on the S input and a logical multiplexer whose S input is connected with the Q output of the flip-flop and whose "1" input is forced to logic "0" and whose "0" input receives said status bit of an addressed memory space, and whose "O" output is connected to said token generator.

5. The Random Access Memory initialization circuit of claim 3, wherein said token generator determines whether all said address spaces are occupied by counting the number of token data allocated against the number of address spaces in said memory.